# **Test Plan (OrangeHRM)**

[**Test Plan (OrangeHRM)** 1](#_Toc182852730)

[Objective 2](#_Toc182852731)

[Scope 2](#_Toc182852732)

[Inclusions 2](#_Toc182852733)

[Test Environments 3](#_Toc182852734)

[Defect Reporting Procedure 3](#_Toc182852735)

[Test Strategy 4](#_Toc182852736)

[Test Schedule 8](#_Toc182852737)

[Test Deliverables 8](#_Toc182852738)

[Entry and Exit Criteria 9](#_Toc182852739)

[Requirement Analysis 9](#_Toc182852740)

[Entry Criteria: 9](#_Toc182852741)

[Exit Criteria: 9](#_Toc182852742)

[Test Execution 9](#_Toc182852743)

[Entry Criteria: 9](#_Toc182852744)

[Exit Criteria: 10](#_Toc182852745)

[Test Closure 10](#_Toc182852746)

[Entry Criteria: 10](#_Toc182852747)

[Exit Criteria: 10](#_Toc182852748)

[Tools 10](#_Toc182852749)

[Risks and Mitigations 10](#_Toc182852750)

[Approvals 11](#_Toc182852751)

#### 

## Objective

objective of this test plan is to validate the functionalities of the **OrangeHRM application (**[**https://opensource-demo.orangehrmlive.com/web/index.php/auth/login**](https://opensource-demo.orangehrmlive.com/web/index.php/auth/login)**)** to ensure it meets the expected quality standards. Testing will focus on verifying core HR management functionalities, usability, and compatibility with specified environments. This includes modules like **Employee Management**, **Leave Management**, and **Time Tracking**.

## Scope

Criteria for success of the testing (number of defects found, time taken for the testing, user satisfaction ratings, etc.)

The roles and responsibilities of the team members involved in the testing: test lead, testers, developers here.

The environments of testing like a different browser, operating system, types of devices.

The timeline and milestones (beginning and end dates) and the planned test activities

Testing tools and equipment to apply, such as testing software, hardware, and testing documentation templates.

## Inclusions

The following items are included in this test plan:

Testing of key OrangeHRM modules:

* **Login**: Validation of user authentication.
* **Employee Management**: Adding, editing, and viewing employee records.
* **Leave Management**: Applying for and approving/rejecting leaves.
* **Time Tracking**: Submission and approval of timesheets.
* **Reports**: Verifying generated reports for accuracy.
* **Admin Features**: Role-based access control and configuration.

**Exclusions:**

* Advanced analytics and integrations (e.g., with payroll systems).
* Load testing for high user volumes (unless explicitly requested).

## Test Environments

The following test environments will be used:

* Development environment
* Test environment
* Production environment

The tests will be conducted on the following environments:

* **Operating System:** Windows 10
* **Browsers:** Google Chrome (primary), Firefox, Microsoft Edge (if possible)

The types of devices and screen sizes that will be used for the testing, including desktop computers, laptops, tablets, and smartphones.

The Network Connectivity and Bandwidth that will be available for testing (Wi-Fi, cellular, and/or wired connections).

Specific hardware or software requirements for executing the test cases (e.g. Particular processor, memory, or storage capacity).

Access credentials such as passwords, tokens, or certificates which will be used to access the test environment which includes the security protocols and authentication mechanism.

Access permissions and roles of the team members who will be using test environment: testers, developers, stakeholders.

Windows 10 – Chrome, Firefox and Edge

• Mac OS – Safari Browser

• Android Mobile OS – Chrome

• iPhone Mobile OS - Safari

## Defect Reporting Procedure

The criteria for identifying a defect, such as deviation from the requirements, user experience issues, or technical errors.

Defects will be reported and tracked in **JIRA**, following a standard format:

* **Description**: Detailed summary of the issue.
* **Steps to Reproduce**: Clear and reproducible steps.
* **Severity and Priority**: Assigned by the tester based on impact.
* **Status**: Updated based on defect progress (Open, In Progress, Closed).
* **Attachments**: Screenshots or other evidence supporting the defect.

## Test Strategy

**Test Types and Approaches**

**1. Functional Testing**

* **Objective**: Validate core HR functionalities like employee record creation and leave workflows.
* **Approach**: Perform end-to-end validation of workflows using predefined test cases. Include both positive (valid inputs) and negative (invalid inputs) scenarios.

**2. Usability Testing**

* **Objective**: Assess the user-friendliness of the interface and navigation.
* **Approach**: Conduct manual usability checks to identify issues in layout, button placement, and navigation within HRM modules.

**3. Compatibility Testing**

* **Objective**: Ensure the application works across supported browsers (Chrome, Firefox).
* **Approach**: Test UI rendering and basic functionality on multiple browsers.

**4. Performance Testing**

* **Objective**: To measure the application’s performance, including page load times and response times, under different usage scenarios.
* **Approach**: Performance testing Tools (Lighthouse for Chrome) will be used to pinpoint and report on slow-loading pages or heavy elements. This is an exploratory type of testing that centres around the possible choke points.

**5. Regression Testing**

* **Objective**: Confirm that recent updates or bug fixes haven’t introduced new issues.
* **Approach**: Re-execute critical test cases in modules like login, leave management, and timesheets.

**5.2 Testing Levels**

**1. Unit Testing**

* **Purpose**: To test individual components or functions in isolation (primarily a developer responsibility).
* **Scope**: We will have unit tests for its important functions, which will be a primary verification before system integration.
* **Objective**: To catch small-scale errors early in development.

**2. Integration Testing**

* **Purpose**: To verify that different modules work together as expected.
* **Scope**: Validate interactions between modules (e.g., employee data reflected in timesheets).
* **Objective**: To catch issues in interactions between distinct functionalities.

**3. System Testing**

* **Purpose**: To validate the complete and integrated software.
* **Scope**: Verify workflows like leave approvals and timesheet submissions.
* **Objective**: To ensure the software meets the business and functional requirements.

**4. Acceptance Testing**

* **Purpose**: To assess the system’s readiness for end-users.
* **Scope**: Acceptance testing will include some high-level functional tests along with some checks on usability to confirm that the application is ready for being showcased to the client.
* **Objective**: To verify that the application meets end-user expectations and requirements.

**5.3 Testing Process Objectives**

1. **Verification of Functional Compliance**: Ensure that each feature meets the specified requirements and performs correctly.
2. **Usability and User Experience Validation**: Confirm that the application provides a smooth, user-friendly experience.
3. **Compatibility Assurance**: Verify that the application works across the intended range of browsers and devices.
4. **Performance Measurement**: Identify any performance issues that could affect the user experience.
5. **Regression Stability**: Ensure that existing functionality is unaffected by new changes.

**5.4 Test Environment**

**Hardware and Software Configurations**:

* **Operating System**: Windows 10
* **Primary Browser**: Chrome (latest stable version)
* **Additional Browsers**: Firefox and Edge (latest stable versions)
* **Tools**: JIRA for defect tracking, Lighthouse for performance assessment.

**Network Requirements**: Tests will be done over regular Wi-Fi with a reliable connection, resembling how someone would commonly use the phone.

**Access Permissions**: Testers will be provided credentials of Demo environment to perform end-to-end tests. You don't need separate access levels for users (customers) and admin roles for front-end testing.

**5.5 Risk Management**

| **Risk** | **Description** | **Mitigation Strategy** |
| --- | --- | --- |
| **Unavailability of Test Environment** | If the demo environment is down, tests cannot proceed as planned. | Have a local backup or mirror of the environment for critical tests. |
| **Incomplete Requirements** | Changes or ambiguities in requirements may impact test accuracy. | Maintain close communication with stakeholders for clarifications. |
| **Test Data Issues** | Lack of diverse test data may affect the comprehensiveness of tests. | Pre-create a range of test data to simulate different user scenarios. |
| **Short Testing Timeline** | Limited time may restrict thorough testing. | Prioritize high-impact test cases and automate where feasible. |

**5.6 Testing Process and Steps**

**Step 1: Test Preparation**

* Review the requirements and finalize test cases.
* Prepare the test data required for specific test scenarios.

**Step 2: Environment Setup**

* Confirm access to the demo site and browser configurations.
* Set up tools such as JIRA for defect logging and Lighthouse for performance testing.

**Step 3: Test Execution**

* Conduct **Smoke Testing** to verify the basic functionality of the application.
* Execute **Functional Testing** for primary user flows.
* Perform **Usability Testing** to identify any interface or user experience issues.

**Step 4: Defect Logging**

* Log all defects found in JIRA with relevant details: steps to reproduce, expected vs. actual results, severity, and priority.
* Attach screenshots and additional documentation for clarity.

**Step 5: Regression Testing**

* After defects are resolved, re-run the test cases around the affected areas to confirm the fixes.
* Execute regression tests on high-priority functionalities to ensure application stability.

**Step 6: Final Reporting**

* Compile results in a **Test Summary Report**, detailing test execution status, defect metrics, and unresolved issues.
* Present findings and receive approvals as per project closure requirements.

## Test Schedule

Following is the test schedule planned for the project –

Task Time Duration

|  |  |
| --- | --- |
| **Task** | **Dates** |
| ▪ Creating Test Plan |  |
| ▪ Test Case Creation |  |
| ▪ Test Case Execution |  |
| ▪ Summary Reports Submission Date |  |

## Test Deliverables

The following are to be delivered to the client.

|  |  |  |
| --- | --- | --- |
| Deliverables | Description | Target Date |
| Test Plan | Details on the scope of the project, test strategy, test schedule, resource requirements, test deliverables and schedule |  |
| Functional Test Cases | Test cases in a detailed manner, covering all the in-scope functionalities |  |
| Defect Report | Defect log with detailed information about defects found with screenshots and steps to reproduce |  |
| Summary Report | Abstract of testing activities, defects, and results. |  |

### Entry and Exit Criteria

The below are the entry and exit criteria for every phase of Software Testing Life

Cycle:

## Requirement Analysis

#### Entry Criteria:

• Once the testing team receives the Requirements Documents or details

about the Project

#### Exit Criteria:

• List of Requirements are explored and understood by the Testing team

• Doubts are cleared

## Test Execution

#### Entry Criteria:

• Test Scenarios and Test Cases Documents are signed-off by the Client

• Application is ready for Testing

#### Exit Criteria:

• Test Case Reports, Defect Reports are ready

## Test Closure

#### Entry Criteria:

• Test Case Reports, Defect Reports are ready

#### Exit Criteria:

• Test Summary Reports

#### Tools

The following are the list of Tools we will be using in this Project:

• JIRA Bug Tracking Tool

• Lighthouse

• Snipping Screenshot Tool

• Word and Excel documents

#### Risks and Mitigations

The following are the list of risks possible and the ways to mitigate them:

|  |  |
| --- | --- |
| Risk | Mitigation |
| Non-Availability of a Resource | Backup Resource Planning |
| Build URL is not working | Resources will work on other tasks |
| Defects delaying testing | Conduct partial testing on other unaffected components. |
| Less time for Testing | Ramp up the resources based on the Client needs dynamically |
| Browser compatibility issues | Focus testing primarily on Chrome, with limited checks on other browsers if resources allow. |

#### Approvals

Team will send different types of documents for Client Approval like below:

• Test Plan

• Test Scenarios

• Test Cases Document

• Reports

Testing will only continue to the next steps once these approvals are done